

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2000-20114

(P2000-20114A)

(43)公開日 平成12年1月21日(2000.1.21)

(51)Int.Cl.⁷

識別記号

F I

テーマコード(参考)

G 0 5 B 19/05

G 0 5 B 19/05

A 5 B 0 7 6

G 0 6 F 9/06

5 3 0

G 0 6 F 9/06

5 3 0 S 5 H 2 2 0

審査請求 未請求 請求項の数2 O L (全 6 頁)

(21)出願番号

特願平10-191418

(22)出願日

平成10年7月7日(1998.7.7)

(71)出願人 000006622

株式会社安川電機

福岡県北九州市八幡西区黒崎城石2番1号

(72)発明者 大庭 賀津男

福岡県北九州市八幡西区黒崎城石2番1号

株式会社安川電機内

(72)発明者 志藤 英己

福岡県北九州市八幡西区黒崎城石2番1号

株式会社安川電機内

Fターム(参考) 5B076 DA06 DF09

5H220 AA04 BB12 CC05 CX02 DD04

DD10 EE08 JJ12 JJ13 JJ42

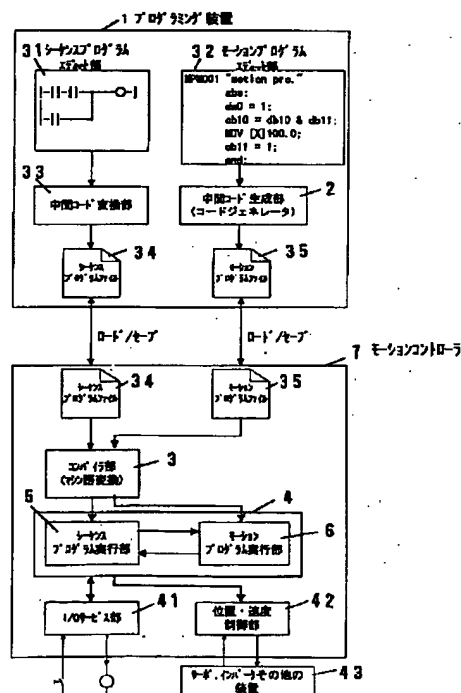
JJ57

(54)【発明の名称】 モーションコントローラのプログラム作成および実行方法

(57)【要約】

【課題】モーションコントローラ側に異なる二つのコンパイルと解析の部品を必要とし、ソフトウェア構造が複雑になること、また、モーションプログラムの逐次もしくは連続(逐次/連続)解析は、シーケンスプログラムの制御スキャンとは非同期であるためモーションとシーケンスの連携的接続時に、互いに解析完了待ちや実行完了待ちの無駄時間を生じ、モーションコントロールの総合的制御周期を高速化出来ないという問題があった。

【解決手段】モーションコントロール言語を、プログラミング装置1上で前記シーケンスコントロール言語と同一形式の中間コードへ変換したのち、モーションコントローラ7へロードするようにし、モーションコントローラ7へロードされたモーションコントロール言語とシーケンスコントロール言語を共通のコンパイラ部3によってスキャン実行型のマシン語へ変換し、スキャン処理部4で実行するようにしたことを特徴とする。



【特許請求の範囲】

【請求項1】 産業機械を制御するモーションコントローラのモーションコントロール言語とシーケンスコントロール言語とをプログラミング装置を用いてそれぞれ独立に記述し、個々に前記モーションコントローラへロードするようにしたモーションコントローラのプログラム作成方法において、前記モーションコントロール言語を、前記プログラミング装置上で前記シーケンスコントロール言語と同一形式の中間コードへ変換したのち、前記モーションコントローラへロードするようにしたことを特徴とするモーションコントローラのプログラム作成方法。

【請求項2】 前記モーションコントローラへロードされたモーションコントロール言語とシーケンスコントロール言語を共通のコンパイラ部によってスキャン実行型のマシン語へ変換し、スキャン処理部で実行するようにしたことを特徴とするモーションコントローラのプログラム実行方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、産業機械を制御するモーションコントローラのプログラム作成および実行方法に関し、とくにモーションプログラムのコード生成およびコンパイル方法に関するものである。

【0002】

【従来の技術】従来の方法を図において説明する。図3は従来例の概念を説明するための図である。図において、30はプログラミング装置、31はシーケンスプログラムエディット部（以下シーケンスプログラムエディタと称する）、32はモーションプログラムエディット部（以下モーションプログラムエディタと称する）、33は中間コード変換部、34はシーケンスプログラムファイル、35はモーションプログラムファイル、36はモーションコントローラ、37はコンパイラ部、38は逐次／連続解析部、39はシーケンスプログラム実行部、40はモーションプログラム実行部、41はI/Oサービス部、42は位置・速度制御部、43はサーボ、インバータその他の装置を表す。モーションプログラムはプログラミング装置30の上ではコード生成（以下コードジェネレートと称する）せずにモーションコントローラ36へロードし、コントローラ側でシーケンスプログラムとは異なるモーションプログラム特有の逐次的または連続的な解析をし、実行する方法をとっていた。図3において、プログラミング装置30にて、モーションプログラムエディタ32とシーケンスプログラムエディタ31を用いてそれぞれのプログラムを個別に表示、作成、編集する。作成後のモーションプログラムは、テキスト形式またはシーケンスプログラムとは異なるデータ形式でモーションプログラムファイル35に保存される。シーケンスプログラムは、中間コード変換部

33を通過してシーケンスプログラムファイル34に保存される。そして、これらファイルはそれぞれ独立にモーションコントローラ36へロードされる。次に、モーションコントローラ36は、プログラムがロードされると、モーションプログラムはそのままのファイル35として保存し、シーケンスプログラムはそれをファイル34として保存するとともにコンパイラ部37を通して実行形式へコンパイルする。この後、CPUが動作（Run）状態の場合は、シーケンスプログラムはスキャン実行方式によりシーケンスプログラム実行部39にて即時運転され、モーションプログラムは、シーケンスプログラム実行部39からのスタート信号を受けて逐次／連続解析部38を通過してイベント実行方式にてモーションプログラム実行部40で運転される。これによりモーションとシーケンスが融合したモーションコントロールが行われる。I/Oサービス部41は、スキャン毎に入力信号の読み込みと出力信号の出力を行う。位置・速度制御部42は、モーションプログラム実行部40で算出された位置・速度の目標値を制御対象であるサーボ、インバータその他の装置43に応じた実際の指令値に変換し出力する。サーボ、インバータその他の装置43はモーションコントローラ36とアナログまたは通信方式により接続される。このように、モーションプログラムとシーケンスプログラムとを別々なコード形式およびファイルとして作成し管理する方法は、シーケンスプログラム中にモーションプログラムを記述し埋め込む方法とは異なり、モーションプログラムの作成と管理の容易性およびプログラムの視認性が高いという利点がある。

【0003】

【発明が解決しようとする課題】しかしながら、上述した従来例においては、図3に示すようにモーションコントローラ側に異なる二つのコンパイルと解析の部品を必要とし、ソフトウェア構造が複雑になること、また、モーションプログラムの逐次もしくは連続（逐次／連続）解析は、シーケンスプログラムの制御スキャンとは非同期であるためモーションとシーケンスの連携の接続時に、互いに解析完了待ちや実行完了待ちの無駄時間を生じ、モーションコントロールの総合的制御周期を高速化出来ないという問題があった。そこで、本発明はモーションプログラムの作成と管理の容易性およびプログラムの視認性が高いという利点を残したまま、コントローラ側のソフトウェア部品（構造）を簡素化し、さらにモーションプログラムとシーケンスプログラムを同一の制御スキャン上で融合し、総合的に高速なモーションコントロールをするためのモーションプログラムの作成および実行方法を提供することを目的とする。

【0004】

【課題を解決するための手段】本発明は、産業機械を制御するモーションコントローラのモーションコントロール言語とシーケンスコントロール言語とをプログラミン

グ装置を用いてそれぞれ独立に記述し、個々に前記モーションコントローラへロードするようにしたモーションコントローラのプログラム作成方法において、前記モーションコントロール言語を、前記プログラミング装置上で前記シーケンスコントロール言語と同一形式の中間コードへ変換したのち、前記モーションコントローラへロードするようにしたことを特徴とし、モーションコントローラへロードされたモーションコントロール言語とシーケンスコントロール言語を共通のコンパイラ部によってスキャン実行型のマシン語へ変換し、スキャン処理部で実行するようにしたことを特徴とする。

【0005】

【発明の実施の形態】以下、本発明の実施例を図に基づいて説明する。図1はプログラミング装置とモーションコントローラの関係を示す概念ブロック図、図2はコードジェネレートの変換ルールの例を示す概念図である。図1において、1はプログラミング装置、2は中間コード生成部（中間コードジェネレータ）、3はコンパイラ部、4はスキャン処理部、5はシーケンスプログラム実行部、6はモーションプログラム実行部、7はモーションコントローラであり、従来と同じものはその説明を省略する。まず、図1で示すプログラミング装置1上に、モーションプログラムエディタ32を用いてモーションプログラムを個別的に表示、作成、編集する。そしてモーションプログラムの保存時にコードジェネレータ2を通してコードジェネレート（中間コードを生成）する。コードジェネレートは、図2に示すように、モーションプログラム20を、ステップ情報パケット22、データパケット24、命令パケット23その他の中間コードデータに分解しモーションプログラムファイル21を生成する。ステップ情報パケット22は、テキスト型・ステップ実行型の命令をスキャン実行型のシーケンスフローチャート構造に変換したもので、モーションプログラム20の場合、①～④の処理BOXとなる。データパケット24は、MOVなどの移動命令に付随するパラメータデータのみを抜き出しテーブル化したものである。命令パケット23は、シーケンスプログラムと同一形式の命令に変換したバイナリーの中間コードであり、スキャン実行される本体である。このように従来の方法に対し、本実施例ではテキスト型・ステップ実行型のモーションプログラムをステップ情報パケット22、データパケット24、命令パケット23の三つのパケットに分解し、スキャン実行型のシーケンスプログラムへ変換するコードジェネレータ2を組み入れたことを特徴とする。次に、モーションプログラムファイル35をモーションコントローラ7へロードする。モーションコントローラ

7は、モーションプログラムがロードされるとシーケンスプログラムと共通のコンパイラ部3を通じてスキャン実行型のマシン語へ変換され、モーションプログラム実行部6で実行される。モーションプログラムの実行は、シーケンスプログラム実行部5からシーケンシャルに接続されておりスキャン処理部4の中の一部として実行される。これにより、モーションとシーケンスの融合が互いの解析待ちや完了待ちという無駄時間なしに実現する。なお、シーケンスプログラムに関してのプログラミング、ロード、コンパイルおよび実行の方法は従来と同じである。

【0006】

【発明の効果】以上述べたように、本発明によれば、モーションプログラムの作成と管理の容易性およびプログラムの視認性が高いという利点を残したまま、コントローラ側のソフトウェア構造を簡素化し、モーションとシーケンスの連携的接続時の無駄時間を取り除き、モーションコントロールの総合的制御周期を高速化できるという効果がある。

【図面の簡単な説明】

【図1】本発明の実施例を示す概念ブロック図

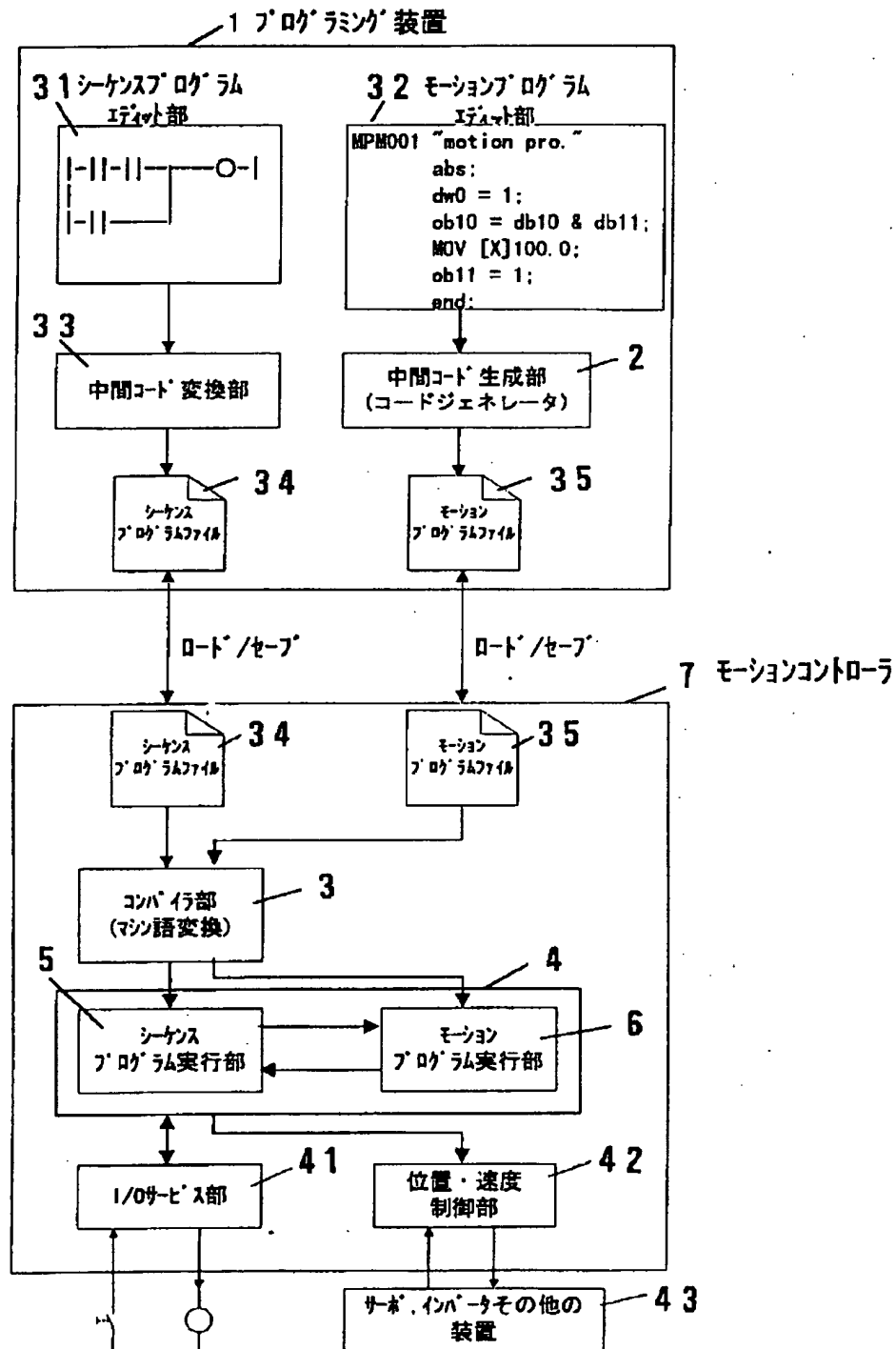
【図2】本発明のコードジェネレータ構成を示す概念図

【図3】従来例の概念ブロック図

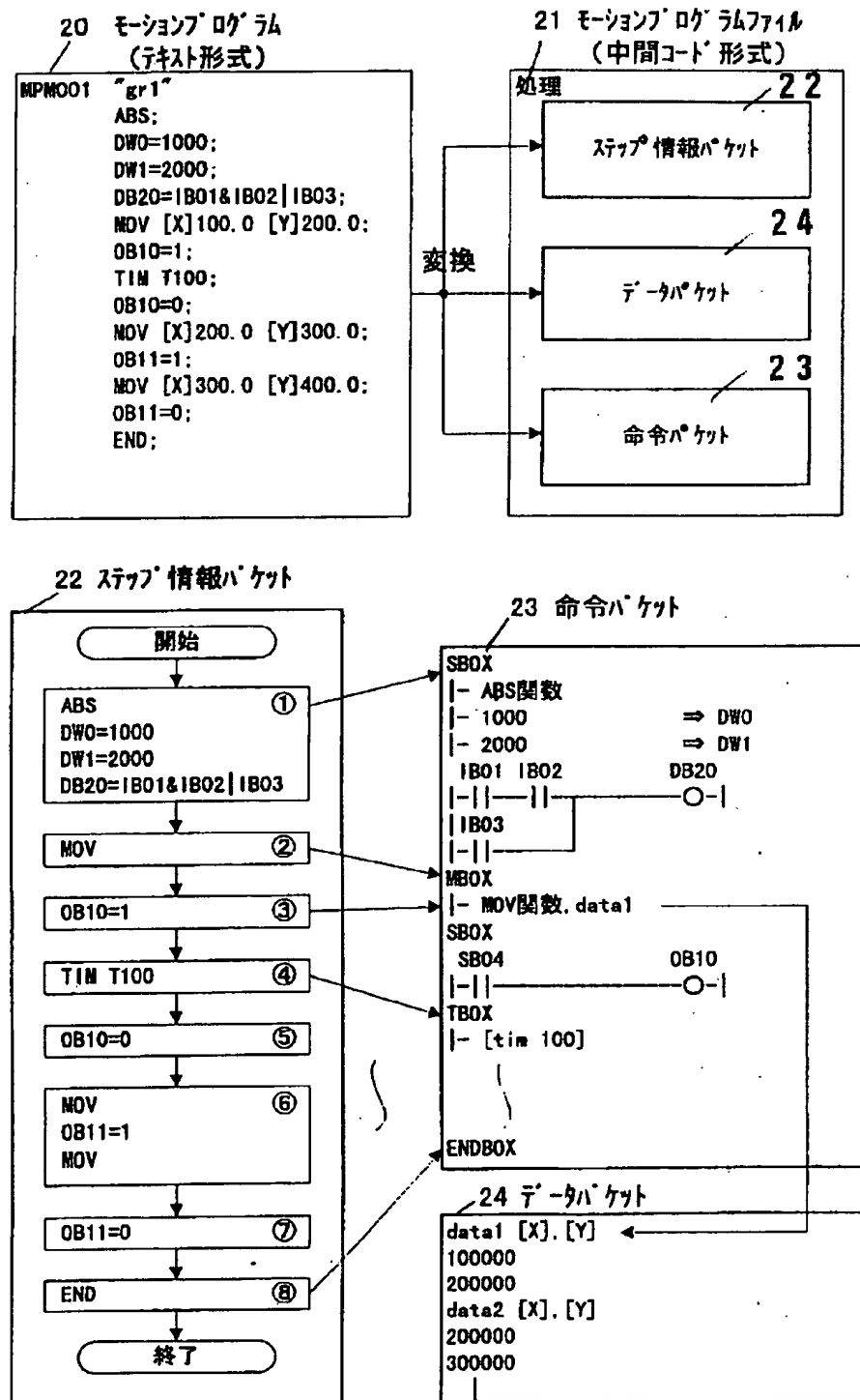
【符号の説明】

- 1 プログラミング装置
- 2 中間コード生成部（中間コードジェネレータ）
- 3、37 コンパイラ部
- 4 スキャン処理部
- 5、39 シーケンスプログラム実行部
- 6、40 モーションプログラム実行部
- 7、36 モーションコントローラ
- 20 モーションプログラム（テキスト形式）
- 21 モーションプログラム（中間コード形式）
- 22 ステップ情報パケット
- 23 命令パケット
- 24 データパケット
- 30 プログラミング装置
- 31 シーケンスプログラムエディット部
- 32 モーションプログラムエディット部
- 33 中間コード変換部
- 34 シーケンスプログラムファイル
- 35 モーションプログラムファイル
- 38 逐次／連続解析部
- 41 I/Oサービス部
- 42 位置・速度制御部
- 43 サーボ、インバータその他の装置

【図1】



【図2】



【図3】

